1. Use a DecisionTreeRegressor to graph **training and test error rates** on various subsets of training data for depths 1, 3, 5, and 10. In other words, you would generate 4 plots, one for each depth. Each plot would show how the training and test error rates change with increasing subsets of training data.

**Depth=1**



**Depth=3**



**Depth=5**



**Depth =10**



**Provide a clear explanation of your interpretation of the graphs.**

Looking into the graphs, we can see that as the training size increases, the training error starting from a lower value increases, and test error starts at a higher level and decreases. This is due to better fit of the model to a certain point of training size.

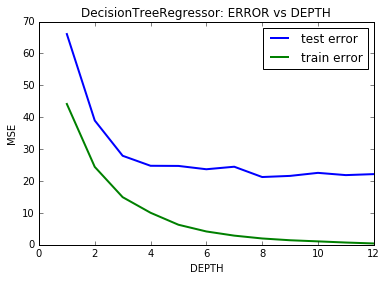
The model predicts better with increasing test size, we can see this from the graphs as the test error is decreasing along with the training size.

We can also observe that for low depths (1,3) the error rate is quite high for both train and test errors which might be due to the poor model built by training (underfit) and the results are biased.

At a depth of 5, we can see that both training and test errors are steadily changing as the training size increases, we can also see that both test and train errors are close together, and hence may be a better fit.

For depth of 10, we can see that the model is overfit, as the train error is close to zero and gap between test and train errors is quite high. Here the model predicts quite well on the training set, but when it comes to testing its not performing well on unseen data.

1. Next, use the DecisionTreeRegressor to graph training and test error rates on varying depths of the tree (use depths 1 to 12). **What is the optimum depth? What is your reason for choosing that number?**



Using different depths, we can see that training and test errors decrease with depth and becomes stable after a certain point. Looking into the graph, optimal depth of 7 or 8 may be ideal because we can see this point indicates a better fit of the model. After a depth of 8 training errors are approaching zero and might imply over fit of the model and will not perform better on unseen data.

1. Finally, use GridSearchCV to identify the best DecisionTreeRegressor estimator using **max\_depth** and two other parameters (e.g., max\_features, splitter). **Predict the home value of a client whose house has the following feature values**:

[[11.95, 0.00, 18.100, 0, 0.6590, 5.6090, 90.00, 1.385, 24, 680.0, 20.20, 332.09, 12.13]]

**Solution:**

Based on grid search, following are the obtained best parameters:

Best param: {'max\_depth': 7, 'max\_features': 13, 'splitter': 'random'}

As estimated before a max\_depth of 7 is ideal depth along with a max\_feature value of 13 and random splitter.

The price of the house for the given sample is predicted at a value 20.76